

PAT-NO: JP410143339A
DOCUMENT-IDENTIFIER: JP 10143339 A
TITLE: NETWORK PRINT SYSTEM

PUBN-DATE: May 29, 1998

INVENTOR-INFORMATION:

NAME	COUNTRY
SATO, MASAYUKI	
SUDO, JUNGO	
HIRAI, NORIO	

ASSIGNEE-INFORMATION:

NAME	COUNTRY
MITSUBISHI ELECTRIC CORP	N/A

APPL-NO: JP08301820

APPL-DATE: November 13, 1996

INT-CL (IPC): G06F003/12 , B41J005/30 , B41J029/38

ABSTRACT:

PROBLEM TO BE SOLVED: To provide a network print system which shortens the time needed for printing even when an abnormality occurs in communications between a client and a server.

SOLUTION: A spooler 12 has a spool file preparation part 15 which divides a print data stream received from a print program 1, prepares a spool file 11 and stores it, a server information management part 18 which manages a server information table 17 which holds transferable and untransferable states, with a server 20, an output destination switching part 19 which switches output destinations when the transfer to the server 20 is not available and a spool file eliminating part 16 which eliminates an unnecessary spool file 11. Every time a spool file 11 is created by the part 15 or each time a spool file is sent front a spool file reinputting part 13 after a fault is recovered, the file 11 is transferred to a spooler 23 of the server 20 and outputted from a printer 21.

COPYRIGHT: (C)1998, JPO

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-143339

(43) 公開日 平成10年(1998) 5月29日

(51) Int.Cl.⁶

識別記号

F I

G 0 6 F 3/12

G 0 6 F 3/12

D

B

B 4 1 J 5/30

B 4 1 J 5/30

Z

29/38

29/38

Z

審査請求 有 請求項の数 9 O L (全 13 頁)

(21) 出願番号 特願平8-301820

(22) 出願日 平成8年(1996)11月13日

(71) 出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 佐藤 雅之

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

(72) 発明者 須藤 純吾

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

(72) 発明者 平井 規郎

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

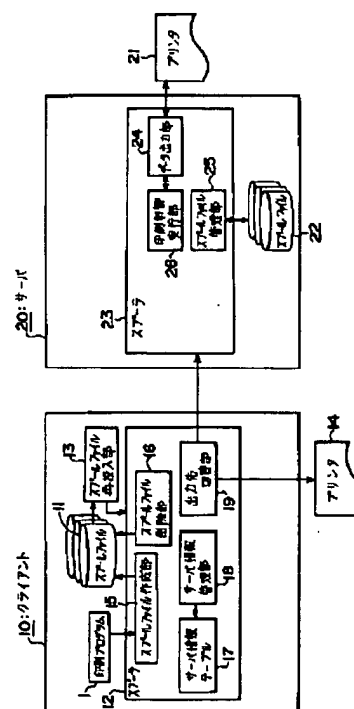
(74) 代理人 弁理士 吉田 研二 (外2名)

(54) 【発明の名称】 ネットワークプリントシステム

(57) 【要約】

【課題】 クライアント・サーバ間の通信で異常が発生した場合でも印刷に要する時間を短縮することのできるネットワークプリントシステムを提供する。

【解決手段】 スプール12は、印刷プログラム1から受け取った印刷データストリームを分割してスプールファイル11として作成し保存するスプールファイル作成部15と、サーバ20との転送可不可状態を保持するサーバ情報テーブル17を管理するサーバ情報管理部18と、サーバ20への転送不可の時出力先を切り替える出力先切替部19と、不要となったスプールファイル11を削除するスプールファイル削除部16と、を有し、スプールファイル11がスプールファイル作成部15により作成される度に若しくは障害復旧後にスプールファイル再投入部13から送られてくる度にスプールファイル11をサーバ20のスプール23に転送し、プリンタ21から出力させる。



【特許請求の範囲】

【請求項1】 印刷プログラムの実行により得られる印刷データをクライアントのスプールからサーバのスプールへ転送し、前記サーバに接続されたプリンタから出力するネットワークプリントシステムにおいて、前記クライアントは、前記印刷プログラムの実行により所定量の印刷データが得られる度にその所定量の印刷データを蓄積するスプールファイルを逐次作成するスプール作成手段を有し、

前記クライアントのスプールは、前記スプール作成手段により前記スプールファイルが作成される度にそのスプールファイルを前記サーバに転送することを特徴とするネットワークプリントシステム。

【請求項2】 前記クライアントは、前記サーバへの転送が正常に終了した前記スプールファイルを削除する印刷データ削除手段を有することを特徴とする請求項1記載のネットワークプリントシステム。

【請求項3】 前記サーバのスプールは、前記クライアントから転送されてくる前記スプールファイルを受け取る度にそのスプールファイルに含まれている印刷データを前記プリンタに出力することを特徴とする請求項1記載のネットワークプリントシステム。

【請求項4】 前記サーバは、前記クライアントから送られてくる一連の前記スプールファイルで構成される印刷データが最後まで正常に受信できなかったときに印刷処理を中止させる印刷中止手段を有することを特徴とする請求項1記載のネットワークプリントシステム。

【請求項5】 前記クライアントは、前記スプールファイルを前記クライアントのスプールに投入する印刷データ再投入手段を有し、

前記クライアントのスプールは、前記印刷データ再投入手段により前記スプールファイルが投入される度にそのスプールファイルを前記サーバに転送することを特徴とする請求項1記載のネットワークプリントシステム。

【請求項6】 前記クライアントは、前記サーバとの転送不可状態を保持するサーバ情報保持手段と、

クライアント・サーバ間の転送不可状態を前記サーバ情報保持手段に設定するサーバ情報管理手段と、を有することを特徴とする請求項1記載のネットワークプリントシステム。

【請求項7】 前記サーバ情報管理手段は、クライアント・サーバ間通信に異常が発生したときに当該サーバとの転送が不可である旨を前記サーバ情報保持手段に設定することを特徴とする請求項6記載のネットワークプリントシステム。

【請求項8】 前記サーバ情報管理手段は、クライアント・サーバ間の通信障害が回復したときに当該サーバとの転送が可能である旨を前記サーバ情報保持手段に設定することを特徴とする請求項6記載のネットワークプリントシステム。

ントシステム。

【請求項9】 前記クライアントは、前記サーバへの転送が不可能なときに前記スプールファイルの転送先を前記サーバのスプールから他へ切り替える転送先切替え手段を有することを特徴とする請求項1記載のネットワークプリントシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、クライアント及びサーバにより構築されたネットワークシステムにおいて、印刷データをクライアントのスプールからサーバのスプールへ転送し、サーバに接続されたプリンタから出力させるネットワークプリントシステムに関する。

【0002】

【従来の技術】図16は、例えば、日経BP社刊Net Wareシステム構築技法に記載された従来のネットワークプリントシステムの全体構成図である。クライアント100は、印刷プログラム101の実行により送られてくる印刷データストリームを受け取りサーバ200へ転送するスプール102を有している。スプール102は、受け取った印刷データストリームを転送可能な形式にデータ変換してスプールファイルとしてを作成するスプール作成部103と、接続したサーバ200に対して印刷データ（スプール）を送出するデータ出力部104とを有している。なお、転送されるスプールは、厳密にはファイルとして作成されるが、そのファイルは、スプール102の内部において便宜上作成されるだけであり外部に保存されることはないため、ファイルではなく単にスプールという表現を用いることにする。

【0003】一方、サーバ200は、プリンタ201と、印刷データを一時保存するスプールファイル202と、クライアント100から送られてくる印刷データを受け取りプリンタ201に出力するスプール203とを有している。スプール203は、印刷データをプリンタ201に出力するデータ出力部204と、印刷データを印刷可能な形式で保存するスプールファイル202を作成するとともに不要となったスプールファイル202を削除するスプールファイル管理部205を有している。

【0004】以上の構成を有する従来例の印刷処理を以下に説明するが、最初に印刷データの転送が正常に終了した場合における処理から説明する。図17に印刷処理が正常に終了したときのシステムタイムチャートを示す。

【0005】印刷プログラム101が実行されることにより作成された印刷データストリームを受け取ると、スプール作成部103は、印刷データストリームからスプールを作成する。スプール102は、作成したスプールをデータ圧縮など所定の転送形式に変換した後にデータ出力部104から出力ポートを介してサーバ200へ転送する。

【0006】サーバ200において、スプールファイル管理部205は、受け取ったスプールを保存するためにスプールファイル202を作成し、印刷可能な形式で保存する。スプーラ203は、スプールファイル202に保存した印刷データの出力処理をスケジューリングし、プリンタ201が未使用であることを確認した後スプールの印刷を開始する。印刷が終了すると、スプールファイル管理部205は、印刷が終了した印刷データを保存するスプールファイル202を削除する。

【0007】次に、クライアント・サーバ間通信に異常が検出されたときの印刷処理について説明する。この異常が検出されたときのシステムタイムチャートを図18に示す。

【0008】正常時の場合と同様に印刷データストリームが作成され、データ出力部104から最初のスプールをサーバ200へ転送しようとしたときに障害が検出されると、その障害が復旧されるまで待ち、障害からの回復後にスプールをサーバ200へ転送する。サーバ200においては、正常時と同様に動作し、受け取ったスプールをスプールファイル202に保存し、その保存した印刷データを出力する。そして、印刷が終了した印刷データを保存するスプールファイル202を削除する。

【0009】また、特開平8-6747号公報には、クライアント側で印刷データと印刷制御情報を蓄積保存する手段を設けた構成が開示されており、この従来例においては、印刷プログラムが作成する印刷データをその蓄積保存する手段に保存した時点で印刷プログラムの処理終了とすることで印刷プログラムの処理速度を短縮でき、また、発生した障害の回復後に保存した印刷データの再転送ができるようにしている。

【0010】

【発明が解決しようとする課題】しかしながら、従来においては、サーバに転送すべき印刷データを全て受け取った後にサーバへの転送処理を行うようにしているため、あまり効率的な印刷処理であるとはいえない。すなわち、例えば、印刷データを作成する処理が1時間要するとした場合、この処理の初期に作成された印刷データは、印刷可能であるのにもかかわらず少なくとも1時間待たされることになる。従って、より効率的にかつ印刷処理に要する時間を短縮できるシステムが望まれる。

【0011】また、クライアント・サーバ間通信に異常が生じているとした場合、従来においては、印刷データの作成後に行うサーバへの転送処理のときになって初めて通信障害を検出することになる。すなわち、例えば、印刷データを作成する処理が1時間要するとした場合、障害の検出は、少なくとも1時間経過した後に初めて気付くことになる。従って、障害が発生したときでも印刷処理に要する時間を短縮できるシステムが望まれる。

【0012】本発明は以上のような問題を解決するためになされたものであり、第1の目的は、サーバ側での印

刷開始をより早く行えるようにすることで印刷に要する時間を短縮することのできるネットワークプリントシステムを提供することにある。

【0013】また、第2の目的は、クライアント・サーバ間の通信で異常が発生した場合でもその障害を迅速に検出することによってサーバ側での印刷開始をより早く行えるようにすることで印刷に要する時間を短縮することのできるネットワークプリントシステムを提供することにある。

【0014】

【課題を解決するための手段】以上のような目的を達成するために、第1の発明に係るネットワークプリントシステムは、印刷プログラムの実行により得られる印刷データをクライアントのスプーラからサーバのスプーラへ転送し、前記サーバに接続されたプリンタから出力するネットワークプリントシステムにおいて、前記クライアントは、前記印刷プログラムの実行により所定量の印刷データが得られる度にその所定量の印刷データを蓄積するスプールファイルを逐次作成するスプール作成手段を有し、前記クライアントのスプーラは、前記スプール作成手段により前記スプールファイルが作成される度にそのスプールファイルを前記サーバに転送するものである。

【0015】第2の発明に係るネットワークプリントシステムは、第1の発明において、前記クライアントは、前記サーバへの転送が正常に終了した前記スプールファイルを削除する印刷データ削除手段を有するものである。

【0016】第3の発明に係るネットワークプリントシステムは、第1の発明において、前記サーバのスプーラは、前記クライアントから転送されてくる前記スプールファイルを受け取る度にそのスプールファイルに含まれている印刷データを前記プリンタに出力するものである。

【0017】第4の発明に係るネットワークプリントシステムは、第1の発明において、前記サーバは、前記クライアントから送られてくる一連の前記スプールファイルで構成される印刷データが最後まで正常に受信できなかったときに印刷処理を中止させる印刷中止手段を有するものである。

【0018】第5の発明に係るネットワークプリントシステムは、第1の発明において、前記クライアントは、前記スプールファイルを前記クライアントのスプーラに投入する印刷データ再投入手段を有し、前記クライアントのスプーラは、前記印刷データ再投入手段により前記スプールファイルが投入される度にそのスプールファイルを前記サーバに転送するものである。

【0019】第6の発明に係るネットワークプリントシステムは、第1の発明において、前記クライアントは、前記サーバとの転送可不可状態を保持するサーバ情報保持手段と、クライアント・サーバ間の転送可不可状態を

前記サーバ情報保持手段に設定するサーバ情報管理手段とを有するものである。

【0020】第7の発明に係るネットワークプリントシステムは、第6の発明において、前記サーバ情報管理手段は、クライアント・サーバ間通信に異常が発生したときに当該サーバとの転送が不可である旨を前記サーバ情報保持手段に設定するものである。

【0021】第8の発明に係るネットワークプリントシステムは、第6の発明において、前記サーバ情報管理手段は、クライアント・サーバ間の通信障害が回復したときに当該サーバとの転送が可能である旨を前記サーバ情報保持手段に設定するものである。

【0022】第9の発明に係るネットワークプリントシステムは、第1の発明において、前記クライアントは、前記サーバへの転送が不可能なときに前記スプールファイルの転送先を前記サーバのスプーラから他へ切り替える転送先切替手段を有するものである。

【0023】

【発明の実施の形態】以下、図面に基づいて、本発明の好適な実施の形態について説明する。

【0024】図1は、本発明に係るネットワークプリントシステムの一実施の形態を示した全体構成図である。図1には、印刷を依頼するクライアント10と印刷を請け負うサーバ20とが接続された構成が示されている。なお、クライアント10は、複数台のサーバ20に接続可能であり、サーバ20は、複数台のクライアント10に接続可能であるが、便宜上1台ずつ図示する。

【0025】クライアント10は、分割された所定量の印刷データを保存するスプールファイル11と、印刷プログラム1の実行により送られてくる印刷データストリームを受け取りサーバ20へ転送するスプーラ12と、スプールファイル11をクライアント10のスプーラ12に投入する印刷データ再投入手段としてのスプールファイル再投入部13と、ローカルに接続されたプリンタ14とを有している。スプーラ12には、スプールファイル作成部15、スプールファイル削除部16、サーバ情報テーブル17、サーバ情報管理部18及び出力先切替部19が含まれている。スプールファイル作成部15は、スプール作成手段として設けられ、印刷プログラム1の実行により所定量の印刷データが得られる度にその所定量の印刷データを蓄積するスプールファイル11を逐次作成する。スプールファイル削除部16は、印刷データ削除手段として設けられ、サーバ20への転送が正常に終了したスプールファイル11を削除する。サーバ情報テーブル17は、サーバ情報保持手段として設けられ、サーバ20との転送不可状態を保持している。図2は、サーバ情報テーブル17の設定内容例を示した図であり、サーバ毎に転送可能フラグを設定する。本実施の形態では、転送可能フラグが“0”のとき転送可能とし、“1”のとき転送不可としている。サーバ情報管理

部18は、サーバ情報管理手段であり、サーバ20への転送不可状態に応じてサーバ情報テーブル17の転送可能フラグの設定を行う。すなわち、クライアント・サーバ間通信に異常が発生したときに転送不可である旨である“1”を、通信障害が回復したときに転送が可能である旨である“0”を、それぞれそのサーバ20に対応した転送可能フラグに設定する。出力先切替部19は、転送先切替手段として設けられ、サーバ20への転送が不可能なときにスプールの転送先をサーバ20のスプーラから他へ切り替える。本実施の形態における出力先切替部19は、図1に示したようにローカルに接続したプリンタ14に出力するようにしている。

【0026】一方、サーバ20は、プリンタ21と、クライアント10から転送されてくるデータを一時保存するために作成されるスプールファイル22と、クライアント10からスプールファイルを受け取る度にプリンタ21に出力するスプーラ23とを有している。スプーラ23には、データ出力部24、スプールファイル管理部25、印刷制御実行部26とが含まれている。データ出力部24は、印刷データをプリンタ21に出力する。スプールファイル管理部25は、印刷可能な形式でスプールファイル22を作成するとともに不要となったスプールファイル22を削除する。印刷制御実行部26は、印刷中止手段として設けられており、クライアント10から送られてくる一連のスプールファイルで構成される印刷データが最後まで正常に受信できなかったときに印刷処理を中止させる。

【0027】なお、上記構成においてスプールファイル11、22は、各計算機10、20に接続されたディスク装置(図示せず)に作成され、その他の構成である印刷プログラム1、スプーラ12、23、スプールファイル再投入部13は、ソフトウェアで作成され、各計算機10、20のCPU上で各機能を発揮することになる。

【0028】本実施の形態において特徴的なことは、クライアント10側で作成された印刷データを所定量毎に分割してスプールファイル11を作成し、スプールファイル11が作成される度にそのスプールファイル11をサーバ20側に転送するようにしたことである。すなわち、クライアント10において印刷データストリームに基づくスプールファイル11の作成処理と、作成されたスプールファイル11のサーバ20への転送処理とを並行して実行できるようにしたことである。これにより、全体を通して印刷処理に要する時間を大幅に短縮することができるようになる。

【0029】次に、本実施の形態における印刷処理について説明する。最初に、新規に印刷データが作成され、その印刷データが正常にサーバ20に転送されプリンタ21から印刷される場合の処理について説明する。なお、図3は、印刷処理が正常に終了したときのタイムチャートを示した図であり、従来例の図17に対応する。

【0030】まず、図4に示したように、印刷プログラム1は、出力先となるサーバ20及びそのサーバ20に接続されているプリンタ21が指定されると、印刷データの作成処理が開始され（ステップ101）、印刷データストリームを作成する（ステップ102）。そして、作成した印刷データストリームをスプール12に順次送出する（ステップ103）。

【0031】図5は、スプール12が行う印刷処理を示したフローチャートである。図5によると、スプール12は、イベントの発生を待ち状態にある（ステップ111）。ここで、印刷プログラム1が実行されたことにより印刷データストリームが送られてくると（ステップ112）、スプールファイル11の作成／転送処理が実行される（ステップ113）。このスプールファイル11の作成／転送処理の詳細について図6に示したフローチャートを用いて説明する。

【0032】まず、この処理を実行するスプールファイル作成／転送ルーチンは、印刷データストリームが発生した場合に呼ばれるルーチンである。まず、変数*i*、*Ret*に1を代入することによって初期化する（ステップ121）。スプールファイル作成部15は、印刷データストリームを逐次受け取ると、所定量の大きさを持つバッファ（図示せず）に読み込む（ステップ122）。そして、バッファサイズ分の読み込んだ印刷データをネットワーク送信が可能な所定の形式に変換し、スプールファイル11として作成して保存する（ステップ123）。このスプールファイル11が1つ作成された時点で図7に示した転送可能チェック処理を行う（ステップ124）。なお、本実施の形態においては、印刷データストリームが分割されるときに所定量は、バッファの大きさに依存することになるが、転送サイズを考慮するなどしてバッファサイズに依存しない任意の所定量を設定したり、マルチバッファを用いたりしてもよい。所定量の設定は、単なる設計事項である。

【0033】転送可能チェックルーチンは、サーバ情報テーブル17の転送可能フラグを参照することによって転送可能かどうかの判断を行う（ステップ131）。本実施の形態では、サーバ情報テーブル17から読み取った値をそのままスプールファイル作成／転送ルーチンに返す（ステップ132、133）。

【0034】図6に示したスプールファイル作成／転送処理において、スプールファイル11の転送が可能な状態（転送可能フラグ＝“0”）であるとき、作成したスプールファイル11をサーバ20へ転送する（ステップ125）。転送が正常に終了したとき、変数*i*をインクリメントする（ステップ126）。

【0035】一方、スプールファイル11の転送が可能な状態でない（転送可能フラグ＝“1”）とき、サーバ情報管理部18に、サーバ情報テーブル17の転送先のサーバ20に対応する転送可能フラグの値に“1”をセ

ットさせる（ステップ127）。そして、本ルーチンのリターンステータス*Ret*に異常を表す“0”を代入し（ステップ128）、変数*i*をインクリメントする（ステップ126）。なお、以上の処理から明かなように、変数*i*は、印刷データストリームの分割数、すなわち作成されるスプールファイル11の総数を算出するための変数である。

【0036】上記処理後、バッファへの読み込みが印刷データストリームの終わりまで達していないときには印刷データストリームのバッファへの読み込み処理（ステップ122）に戻る（ステップ129）。印刷データストリームの終わりに達したとき、グローバル変数*ALL*に（*i*－1）を代入してリターンステータス*Ret*を返す（ステップ130）。なお、グローバル変数*ALL*の値は、印刷データストリームを分割することにより作成されたスプールファイルの総数である。

【0037】以上のようにして、本実施の形態によれば、図5に示したスプールファイル作成／転送処理（ステップ113）において、作成された印刷データストリームに基づいてスプールファイル11が作成される度にサーバ20に逐次転送することになる。すなわち、図3から明かなように、本実施の形態では、バッファが印刷データストリームの一部で満たされスプールファイル11が作成される時間のみ待つだけでサーバ20への転送処理が開始されるので、印刷処理時間を大幅に短縮することができる。

【0038】図5において、スプールファイル11の転送／作成処理が、異常終了したとき（*Ret*＝0）には、イベントの待ち状態に戻り（ステップ111）、正常終了したとき（*Ret*＝1）には、スプールファイル11の削除を行うが（ステップ114）、このステップ114におけるスプールファイル削除部16が行うスプールファイル削除処理について図8に示したフローチャートを用いて説明する。

【0039】スプールファイル削除ルーチンでは、変数*i*を0で初期化した後（ステップ141）、スプールファイル（*i*）を削除する（ステップ142）。変数*i*をインクリメントし（ステップ143）、グローバル変数*ALL*に変数*i*が達するまで上記処理を繰り返す（ステップ144）。これにより、クライアント10において作成された全スプールファイル11を削除することができる。

【0040】次に、スプールファイル11が転送されてくるサーバ20における印刷処理について図9を用いて説明する。まず、クライアント10から一連のスプールファイルを受け取れたときの正常時における処理から説明する。

【0041】サーバ20におけるスプール23は、スプールファイルを受け取ると、それをサーバ20側のスプールファイル22として保存する（ステップ151）。

そして、スプール23は、最初のスプールファイルを受け取ったときにプリンタ21が印刷要求待ち状態であればスプールファイル22に含まれている印刷データの印刷を開始する(ステップ152)。印刷要求待ち以外の状態であれば、プリンタ21への出力処理をスケジューリングしておく。そして、クライアント10から最終のスプールファイルが転送されてくるまで上記処理を繰り返す(ステップ154)。スプール23が最終のスプールファイルを受け取り、プリンタ21が印刷を正常に終了すると、その旨をクライアント10に通知する(ステップ155)。

【0042】ところで、本実施の形態では、1つの印刷データストリームを複数のスプールファイル11に分割してサーバ20に転送するようにしているが、各スプールファイル11には、後続するスプールファイル11の存否を表す情報が含まれている。従って、サーバ20のスプール23は、この情報をスプールファイルの受信の度に確認することによって後続するスプールファイルの存否及び最後尾であるかどうかを判断することができる。これにより、ステップ152において、本来後続して転送されてくるはずのスプールファイルが所定時間内に受信しなかったときは、何らかの通信障害が発生したとして印刷を中止することになる(ステップ156)。ここで、印刷制御実行部26が行う印刷を中止する処理の詳細を図10を用いて説明する。

【0043】印刷制御実行部26は、タイムアウトの発生時、受信できなかったスプールファイルと同じ印刷データストリームを構成する他のスプールファイル22を印刷しているかどうかを調べる(ステップ161)。印刷中のものがあれば、プリンタ21に印刷中止コードを送出し、強制的に印刷を中止させる(ステップ162)。一方、印刷中のものがなければ、既に受け取っている他のスプールファイル22は、まだ印刷されていないと判断してスケジュールから削除する(ステップ163)。その後、1つの印刷データストリームを構成する一連のスプールファイル22を全て削除する(ステップ164)。異常終了のため、正常に受け取った一連のスプールファイル22を保持していても無駄だからである。但し、一連のスプールファイル22に通し番号が付されていて、サーバ20からクライアント10に再送して欲しい通し番号を通知するなどすれば、受信済みのスプールファイル22を削除しない方が効率的である。このようにして、クライアント・サーバ間通信に障害が発生した場合にもサーバ20において対処することができる。

【0044】本実施の形態では、以上のように所定量の印刷データストリームが作成される度にサーバ20にスプールファイルを送るようにした。図3と図17とを比較すると、クライアント10側におけるジョブのみならず、印刷処理に要する時間を大幅に短縮できたことが一

目瞭然である。もちろん、印刷データの作成、データ転送、受信、印刷それぞれに要する時間は、従来例と同様である。本実施の形態においては、印刷データを分割してスプールファイルを作成することによってクライアント10におけるスプールファイルの作成処理と転送処理とを並行してできるようにしたので、印刷処理に要する時間を大幅に短縮することができる。

【0045】以上の処理は、基本的に印刷データが正常にサーバ20に転送されプリンタ21から印刷される場合の処理について説明したが、次に、作成したスプールファイル11がサーバ20に転送されなかったときの異常時における処理について説明する。なお、図11は、クライアント・サーバ間の転送処理に異常を検出したときのシステムタイムチャートを示した図であり、従来例の図18に対応する。

【0046】図5において、スプール12は、イベントの発生の待ち状態にあるときに印刷データストリームが送られてくると、スプールの作成/転送処理が実行される(ステップ111~113)。このスプールの作成/転送処理の詳細については、図6に示したフローチャートを用いてすでに説明した。ところで、図6におけるステップ124、つまり図7に示した転送可能チェック処理により指定のサーバ20との通信が可能であるかどうかを判断しているが、この転送可能チェック処理は、最初のスプールファイル11が作成された時点で実行される。すなわち、従来においては、印刷データストリーム全体が作成されるのを待って初めて転送先となるサーバ20との通信が可能かどうかの判断をしていたのに対し、本実施の形態においては、印刷データストリームの一部すなわち最初のスプールファイル11が作成され転送しようとした時点でその判断を行うことができる。

【0047】以上のように、サーバ20への転送ができない状態にあるとき、すなわちステップ113におけるスプール作成/転送処理からのリターンステータスが異常終了を表す“0”であったとき、イベントの発生の待ち状態に戻るが(ステップ111)、転送が実行できない間にも残りのスプールファイル11の作成処理は、継続して行われる。

【0048】ところで、サーバ情報管理部18は、自らサーバ間の通信状態を定期的に監視し、又は間接的にシステム管理者等による設定入力により障害が復旧されたことを検出すると、当該サーバ20に対応したサーバ情報テーブル17の転送可能フラグを“0”にセットする。図12にスプールファイル再投入部13における処理のフローチャートを示すが、スプールファイル再投入部13は、システム管理者等により起動され、又は障害時にサーバ情報テーブル17の転送可能フラグを常時監視し、当該サーバ20の転送可能フラグが“0”にセットされたことを検出すると、通信が復旧したと認識し(ステップ171)、スプール12に作成されているス

11

プールファイル11を順次送出する(ステップ172)。

【0049】図5において、イベント待ちしていたスプール12は、スプールファイル再投入部13からスプールファイル11を受け付けると(ステップ111、112、115)、スプールファイル転送処理を行う(ステップ116)。この処理は、図6に示したスプールファイル作成/転送処理からスプールファイル作成処理を排除したのとはほぼ同様になる。このスプールファイル転送処理のフローチャートを図13に示す。

【0050】図13において、スプール12は、まず、変数*i*、*Ret*に1を代入することによって初期化する(ステップ181)。そして、作成されているスプールファイル11をサーバ20へ転送する(ステップ182)。転送が正常に終了したときには、変数*i*をインクリメントする(ステップ183)。この処理を変数*i*がグローバル変数ALLに達するまで上記処理を繰り返し行う(ステップ184)。

【0051】一方、スプールファイル11の転送が失敗したとき、サーバ情報管理部18に、サーバ情報テーブル17の転送先のサーバ20に対応する転送可能フラグの値に“1”をセットさせる(ステップ185)。そして、本ルーチンのリターンステータス*Ret*に異常を表す“0”を代入する(ステップ186)。そして、スプールファイル11の転送が正常に実行でき印刷データストリームの末尾に達したとき、あるいはスプールファイル11の転送が正常に実行できなかった時点でリターンステータス*Ret*を返し(ステップ187)、本処理を終了する。

【0052】以上のように、本実施の形態によれば、印刷データストリームをスプールファイル11に保存するようにしたので、障害から復旧した時点で印刷プログラム1を再起動しなくても作成されたスプールファイル11をサーバ20に転送することができる。そして、図5におけるスプールファイル転送処理(ステップ116)の後、前述したスプールファイル削除処理(ステップ114)を行う。

【0053】図11から明らかなように、本実施の形態では、1つのスプールファイル11の作成完了後にサーバ20への転送処理を開始するようにしているので、その時点でクライアント・サーバ間通信の異常を検出することができる。これにより、印刷データストリームからスプールファイル11全ての作成を待たずにそのときからすぐに障害の復旧作業に取りかかることができる。また、その時間分、サーバ20への転送処理も早く再開することができるので、印刷処理に要する時間を大幅に短縮することができる。

【0054】ところで、サーバ情報管理部18がサーバ情報テーブル17の設定値を変更することは、上述した通りであるが、このサーバ情報管理部18において行わ

12

れる処理について図14に示したフローチャートを用いて簡単に説明すると、サーバ情報管理部18は、サーバ名と更新すべき転送可能フラグの値を受け付けると(ステップ191)、サーバ情報テーブル17の指定されたサーバ20の転送可能フラグに指定の転送可能フラグ値をセットする(ステップ192)。このようにして、クライアント10が接続されている各サーバへの転送が可能かどうかをサーバ情報テーブル17に設定するようにしたので、その判断を容易に行うことができる。

10 【0055】また、本実施の形態では、出力先切替部19を設けたことも特徴の一つとしている。この出力先切替部19における処理について図15に示したフローチャートを用いて説明する。

【0056】出力先切替部19は、スプールファイル11の出力先すなわちサーバ名を受け付けると(ステップ201)、サーバ情報テーブル17のそのサーバ20に対応した転送可能フラグの値をサーバ情報テーブル17から受け取る(ステップ202)。その値が転送可能(転送可能フラグ=“0”)である場合、出力先をサーバ20に向ける(ステップ203)。転送不可(転送可能フラグ=“1”)である場合、出力先をプリンタ14が接続されているローカルポートに向ける(ステップ204)。これにより、サーバ20に接続されているプリンタ21からの出力が不可能でときであってもローカルに接続した代わりのプリンタ14から印刷データを出力させることができる。

20 【0057】なお、本実施の形態では、サーバ20への転送が不可能なときには無条件にローカルのプリンタ14へ出力を切り替えるようにしたが、ローカルのプリンタ14でなくても予備的な他のサーバへ切り替えるようにしてもよい。また、切替え先を設定したテーブルなどを持つことで、任意のサーバへ出力先を切り替えるようにすることもできる。

【0058】

【発明の効果】本発明によれば、クライアント側で作成された印刷データを所定量毎に分割してスプールファイルを作成し、スプールファイルが作成される度にそのスプールファイルをサーバ側に転送するようにした。すなわち、スプールファイルが作成される時間のみ待てば、サーバへの転送処理が開始され、その後は、印刷データに基づくスプールファイルの作成処理と、作成されたスプールファイルのサーバへの転送処理とを並行して実行できるようにしたので、印刷処理全体に要する時間を大幅に短縮することができる。

【0059】また、印刷データ削除手段を設けたので、転送が正常に終了したスプールファイルを削除することができる。

【0060】また、サーバ側において、転送されてくるスプールファイルを受け取ると即座に印刷を開始するようにしたので、スプールファイルの受信処理とプリンタ

への出力処理とを並行して行うことができる。これにより、印刷処理全体に要する時間を大幅に短縮することができる。

【0061】また、サーバ側において、転送されてくるはずのスプールファイルを正常に受け取ることができなかったとき、そのスプールファイルと同じ印刷データストリームを構成する他のスプールファイルが印刷途中であればそれを中止し、まだ印刷されていないときはスケジュールから削除するようにしたので、無駄な印刷をしなくてすむ。

【0062】また、印刷データ再投入手段を設けたことにより、障害が発生しその復旧をした後に、作成され保存されたままのスプールファイルの転送を再開することができる。

【0063】また、サーバ情報管理手段を設け、クライアントが接続されている各サーバへの転送が可能かどうかをサーバ情報保持手段に設定するようにしたので、転送可否の判断を容易に行うことができる。

【0064】また、転送先切替え手段を設けたので、本来印刷を行うべきサーバとの通信が正常にできない場合であっても他のサーバやローカルのプリンタへその出力先を切り替えることができる。

【図面の簡単な説明】

【図1】 本発明に係るネットワークプリントシステムの一実施の形態を示した全体構成図である。

【図2】 本実施の形態におけるサーバ情報テーブルの設定内容例を示した図である。

【図3】 本実施の形態において印刷処理が正常に終了したときのタイムチャートを示した図である。

【図4】 本実施の形態における印刷プログラムにおける処理を示したフローチャートである。

【図5】 本実施の形態におけるクライアントのスプーラが行う印刷処理を示したフローチャートである。

【図6】 図5に示したスプールファイル作成／転送処理の詳細を示したフローチャートである。

【図7】 図6に示した転送可能チェック処理を示した

フローチャートである。

【図8】 図5に示したスプールファイル削除部が行うスプールファイル削除処理を示したフローチャートである。

【図9】 本実施の形態におけるサーバのスプーラが行う印刷処理を示したフローチャートである。

【図10】 図9に示した印刷制御実行部が行う印刷中止処理を示したフローチャートである。

【図11】 本実施の形態においてクライアント・サーバ間の転送処理に異常が検出されたときのシステムタイムチャートを示した図である。

【図12】 本実施の形態のスプールファイル再投入部における処理を示したフローチャートである。

【図13】 図5に示したスプールファイル転送処理を示したフローチャートである。

【図14】 本実施の形態のサーバ情報管理部において行われる処理を示したフローチャートである。

【図15】 本実施の形態の出力先切替部において行われる処理を示したフローチャートである。

【図16】 従来のネットワークプリントシステムを示した全体構成図である。

【図17】 従来において印刷処理が正常に終了したときのタイムチャートを示した図である。

【図18】 従来においてクライアント・サーバ間の転送処理に異常が検出されたときのシステムタイムチャートを示した図である。

【符号の説明】

1 印刷プログラム、10 クライアント、11、22 スプールファイル、12、23 スプーラ、13 スプールファイル再投入部、14、21 プリンタ、15 スプールファイル作成部、16 スプールファイル削除部、17 サーバ情報テーブル、18 サーバ情報管理部、19 出力先切替部、20 サーバ、24 データ出力部、25 スプールファイル管理部、26 印刷制御実行部。

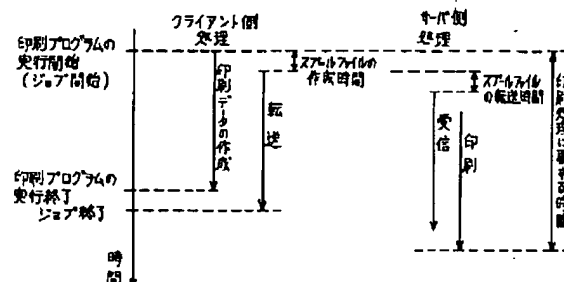
【図2】

サーバ情報テーブル

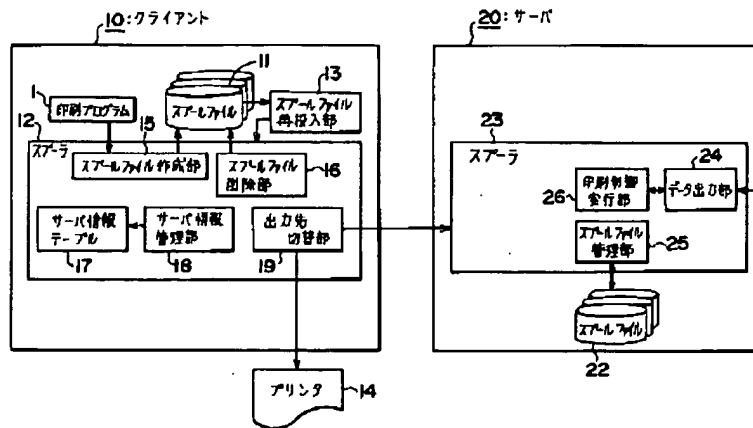
サーバ名	転送可能フラグ
サーバ1	1
サーバ2	0
サーバ3	1

【図3】

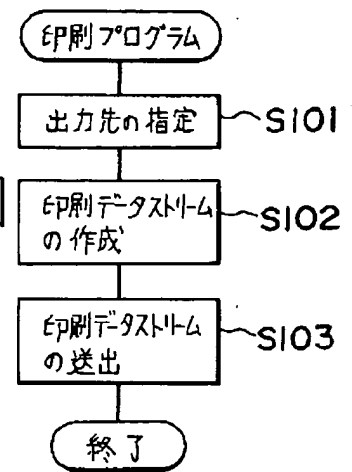
本実施の形態における正常時のシステムタイムチャート



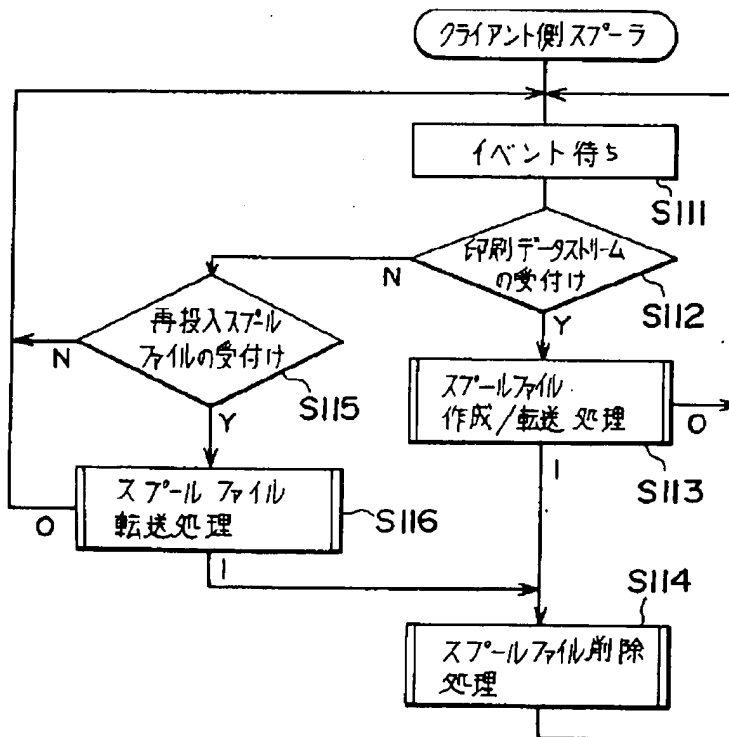
【図1】



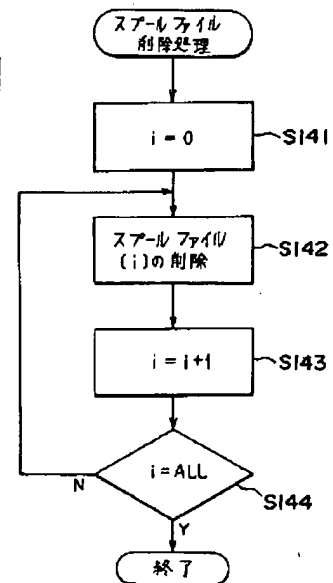
【図4】



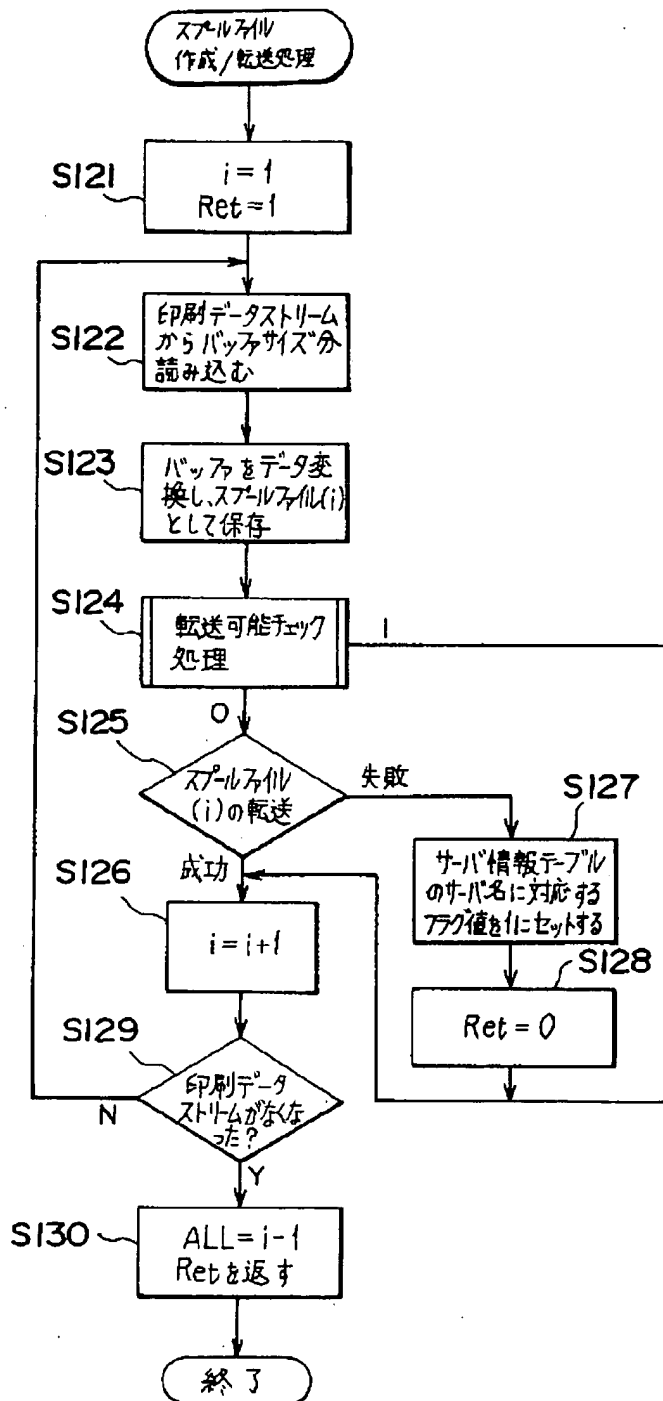
【図5】



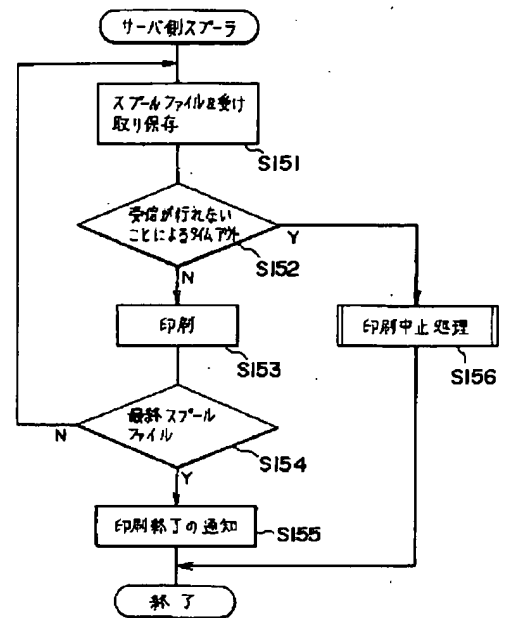
【図8】



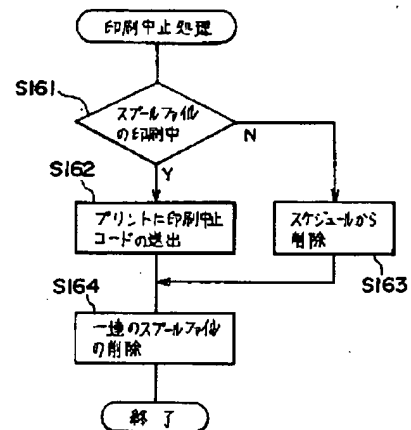
【図6】



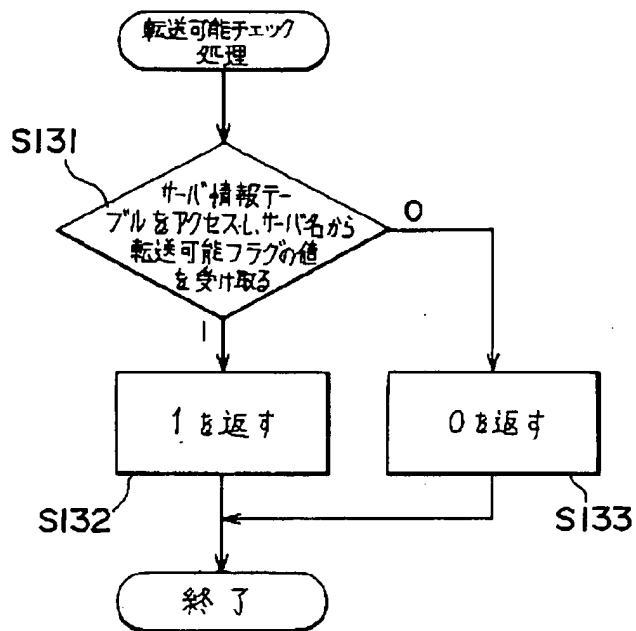
【図9】



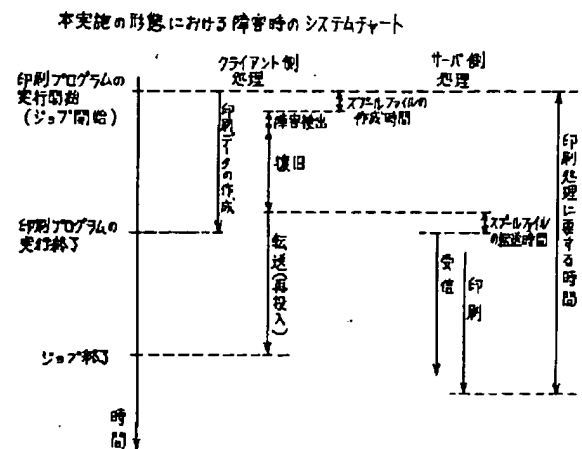
【図10】



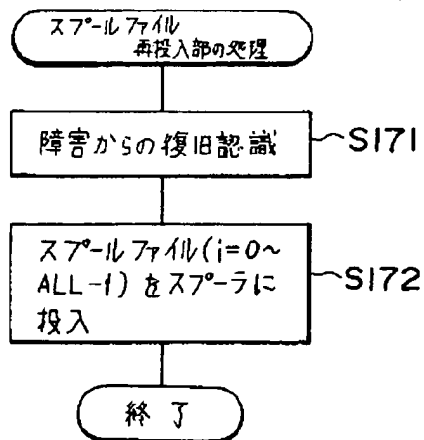
【図7】



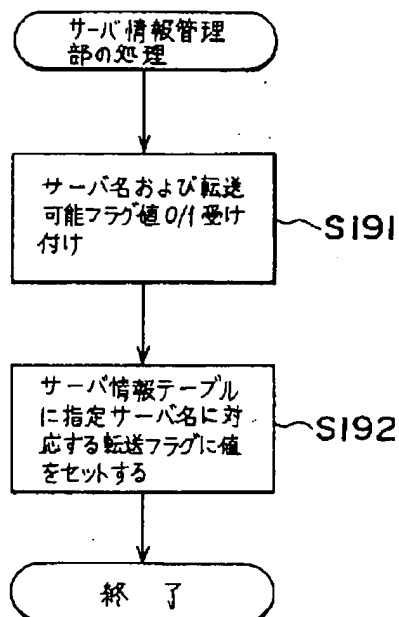
【図11】



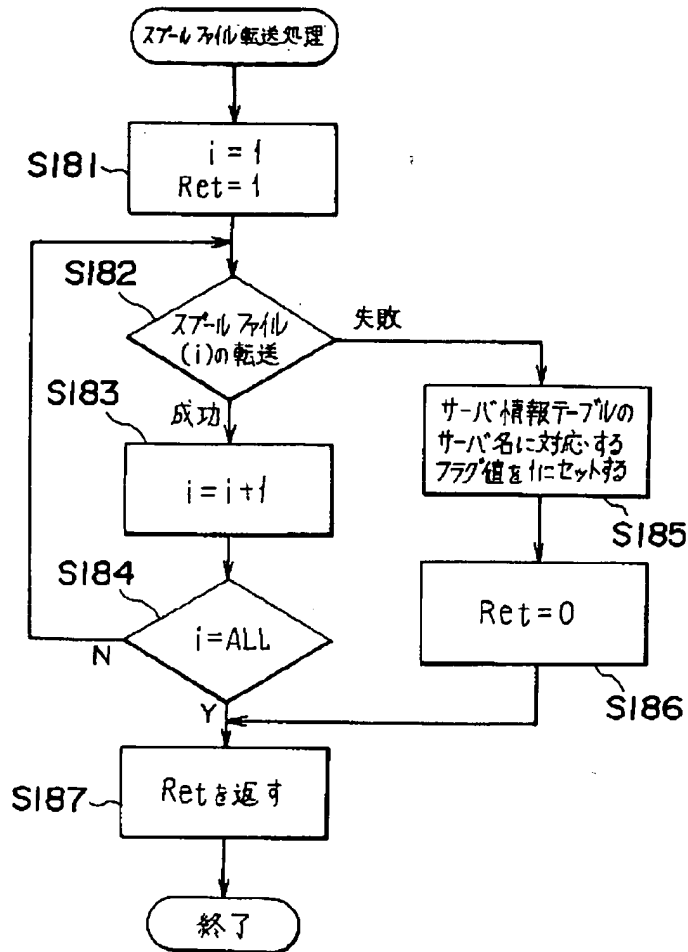
【図12】



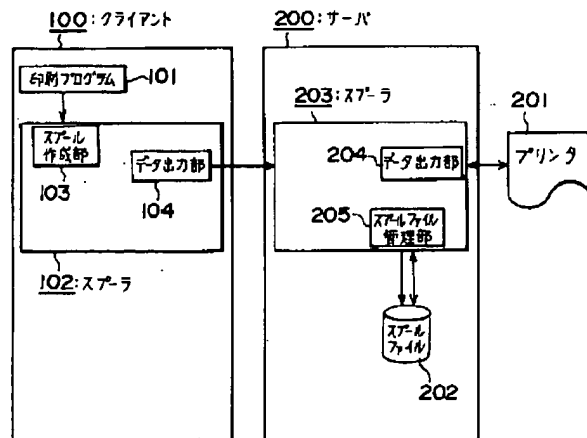
【図14】



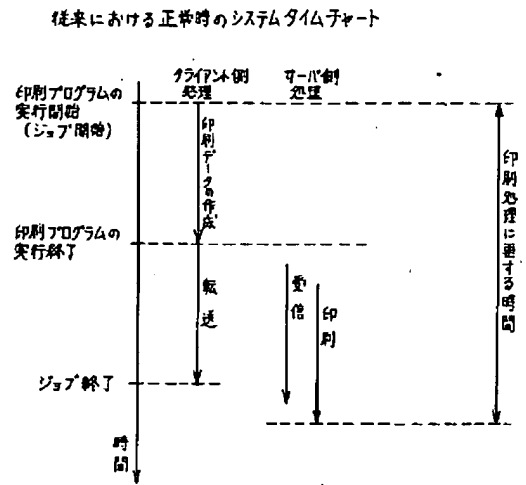
【図13】



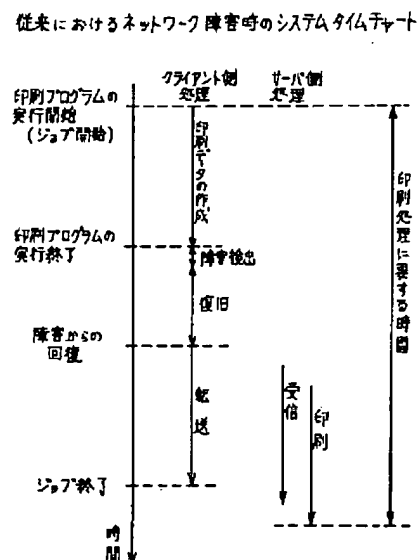
【図16】



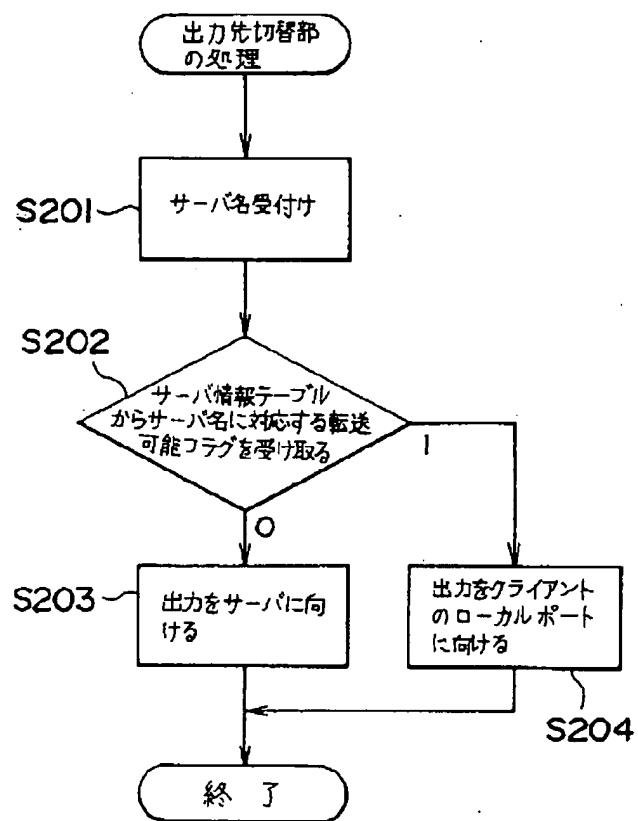
【図17】



【図18】



【図15】



* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] In the network system built by the client and the server, this invention transmits print data to the spooler of a server from the spooler of KURAIINTO, and relates to the network printing system made to output from the printer connected to the server.

[0002]

[Description of the Prior Art] Drawing 16 is the conventional network printing system whole block diagram indicated by for example, the Nikkei Business Publications Co. ** NetWare SYSTEM construction technique. The client 100 has the spooler 102 which transmits the printing data stream sent by activation of the printing program 101 to the reception server 200. The spooler 102 has the spool creation section 103 which carries out data conversion to the format that the received printing data stream can be transmitted, and creates as a spool file, and the data output section 104 which sends out print data (spool) to the connected server 200. In addition, although the spool transmitted is strictly created as a file, since the file is only created for convenience in the interior of a spooler 102 and is not saved outside, it is carried out to using not a file but an expression only called a spool.

[0003] On the other hand, the server 200 has the printer 201, the spool file 202 which saves print data temporarily, and the spooler 203 which outputs the print data sent from a client 100 to the reception printer 201. A spooler 203 has the spool file Management Department 205 which deletes the spool file 202 which became unnecessary while creating the spool file 202 saved in the data output section 204 which outputs print data to a printer 201, and the format which can print print data.

[0004] Although printing processing of the conventional example in which it has the above configuration is explained below, it explains from processing when a transfer of print data is completed normally first. A system time chart when printing processing is normally completed to drawing 17 is shown.

[0005] If the printing data stream created by performing the printing program 101 is received, the spool creation section 103 will create a spool from a printing data stream. After a spooler 102 changes the created spool into predetermined transfer formats, such as a data compression, it is transmitted to a server 200 through an output port from the data output section 104.

[0006] In a server 200, since the received spool is saved, the spool file Management Department 205 creates a spool file 202, and it saves in the format which can be printed. A spooler 203 carries out scheduling of the output processing of the print data saved at the spool file 202, and after it checks that a printer 201 is intact, it starts printing of a spool. After printing is

completed, the spool file Management Department 205 deletes the spool file 202 which saves the print data which printing ended.

[0007] Next, printing processing when abnormalities are detected by the communication link between client-server is explained. A system time chart when this abnormality is detected is shown in drawing 18.

[0008] A printing data stream is created like the case of forward always, and if a failure is detected when it is going to transmit the first spool to a server 200 from the data output section 104, after recovering from waiting and a failure, a spool will be transmitted to a server 200, until the failure is restored. In a server 200, the spool which operated like always [forward] and was received is saved at a spool file 202, and the saved print data are outputted. And the spool file 202 which saves the print data which printing ended is deleted.

[0009] Moreover, it is made the re-transfer of the print data saved after recovery of the failure the configuration which established the means which carries out are-recording preservation of print data and the printing control information was indicated by JP,8-6747,A by the client side, the processing speed of a printing program could be shortened by carrying out as processing termination of a printing program when the print data which a printing program creates saved in this conventional example to that means that carries out are-recording preservation, and generated being able to do.

[0010]

[Problem(s) to be Solved by the Invention] However, in the former, in order to perform transfer processing to a server after receiving all the print data that should be transmitted to a server, it cannot be said that it is not much efficient printing processing. That is, for example, when [the processing which creates print data] requiring for 1 hour, the print data created in early stages of this processing will be kept waiting for at least 1 hour, in spite of being able to print. Therefore, a system which can shorten the time amount which printing processing takes more efficiently is desired.

[0011] Moreover, in the former, when abnormalities have arisen in the communication link between client-server, communication failure will be detected only after becoming at the time of the transfer processing to the server performed after creation of print data. That is, for example, when [the processing which creates print data] requiring for 1 hour, detection of a failure will be noticed for the first time, after at least 1 hour passes. Therefore, even when a failure occurs, a system which can shorten the time amount which printing processing takes is desired.

[0012] Made in order that this invention may solve the above problems, the 1st object is to offer the network printing system which can shorten the time amount which printing takes by enabling it to perform more printing initiation by the side of a server early.

[0013] Moreover, the 2nd object is by detecting the failure promptly to offer the network printing system which can shorten the time amount which printing takes by enabling it to perform more printing initiation by the side of a server early, even when abnormalities occur in the communication link between client-server.

[0014]

[Means for Solving the Problem] In order to attain the above objects, the network printing system concerning the 1st invention In the network printing system outputted from the printer which transmitted the print data obtained by printing program execution to the spooler of a server from the spooler of KURAIMTO, and was connected to said server Said client has a spool creation

means to create serially the spool file which accumulates the print data of the specified quantity whenever the print data of the specified quantity are obtained by said printing program execution. The spooler of said client transmits the spool file to said server, whenever said spool file is created by said spool creation means.

[0015] In the 1st invention, as for the network printing system concerning the 2nd invention, said client has a print-data deletion means to delete said spool file which the transfer to said server ended normally.

[0016] The network printing system concerning the 3rd invention outputs the print data contained in the spool file whenever the spooler of said server receives said spool file transmitted from said client to said printer in the 1st invention.

[0017] It has a printing termination means by which the network printing system concerning the 4th invention stops printing processing when the print data which consist of said spool files of a single string sent from said client are not able to receive said server to normal to the last in the 1st invention.

[0018] Said client has a print-data reclosing means by which the network printing system concerning the 5th invention supplies said spool file to the spooler of said KURAIINTO in the 1st invention, and the spooler of said client transmits the spool file to said server, whenever said spool file is supplied by said print-data reclosing means.

[0019] In the 1st invention, as for the network printing system concerning the 6th invention, said client has a server information maintenance means to hold a transfer good improper condition with said server, and a server information management means to set the transfer good improper condition between client-server as said server information maintenance means.

[0020] In the 6th invention, when abnormalities generate said server information management means in the communication link between client-server, as for the network printing system concerning the 7th invention, the purport for which a transfer with the server concerned is improper is set as said server information maintenance means.

[0021] In the 6th invention, the network printing system concerning the 8th invention sets the purport in which a transfer with the server concerned is possible as said server information maintenance means, when the communication failure between client-server recovers said server information management means.

[0022] Said client has a destination change means by which the network printing system concerning the 9th invention changes the destination of said spool file from the spooler of said server to others when the transfer to said server is impossible in the 1st invention.

[0023]

[Embodiment of the Invention] Hereafter, the gestalt of suitable operation of this invention is explained based on a drawing.

[0024] Drawing 1 is the whole block diagram having shown the gestalt of 1 operation of the network printing system concerning this invention. The configuration in which the client 10 which requests printing, and the server 20 which contracts printing were connected to drawing 1 is shown. In addition, it is connectable with two or more sets of servers 20, and a client 10 illustrates one server 20 at a time for convenience, although it is connectable with two or more sets of clients 10.

[0025] The client 10 has the printer 14 locally connected with the spool file 11 which saves the divided print data of the specified quantity, the spooler 12 which transmits the printing data stream sent by activation of the printing

program 1 to the reception server 20, and the spool file reclosing section 13 as a print-data reclosing means to supply a spool file 11 to the spooler 12 of KURAI10 10. The spool file creation section 15, the spool file cutout 16, the server information table 17, the server Research and Data Processing Department 18, and the output destination change change section 19 are contained in the spooler 12. The spool file creation section 15 is formed as a spool creation means, and creates serially the spool file 11 which accumulates the print data of the specified quantity whenever the print data of the specified quantity are obtained by activation of the printing program 1. The spool file cutout 16 is formed as a print-data deletion means, and deletes the spool file 11 which the transfer to a server 20 ended normally. The server information table 17 is formed as a server information maintenance means, and holds the transfer good improper condition with a server 20. Drawing 2 is drawing having shown the example of the content of setting out of the server information table 17, and sets up the flag which can be transmitted for every server. With the gestalt of this operation, when the flag which can be transmitted is "0", a transfer is made possible, and when it is "1", it is supposed that a transfer is impossible. The server Research and Data Processing Department 18 is a server information management means, and sets up the flag of the server information table 17 which can be transmitted according to the transfer good improper condition to a server 20. That is, "0" which is the purport which can be transmitted when communication failure recovers "1" which is a purport [that it cannot transmit when abnormalities occur in the communication link between client-server] is set as the flag corresponding to the server 20 which can be transmitted, respectively. The output destination change change section 19 is formed as a destination change means, and when the transfer to a server 20 is impossible, it changes the destination of a spool from the spooler of a server 20 to others. He is trying to output the output destination change change section 19 in the gestalt of this operation to the printer 14 locally connected as shown in drawing 1 .

[0026] On the other hand, the server 20 has the printer 21, the spool file 22 created since the data transmitted from a client 10 are saved temporarily, and the spooler 23 outputted to a printer 21 whenever it receives a spool file from a client 10. The data output section 24, the spool file Management Department 25, and the printing control activation section 26 are contained in the spooler 23. The data output section 24 outputs print data to a printer 21. The spool file Management Department 25 deletes the spool file 22 which became unnecessary while creating a spool file 22 in the format which can be printed. The printing control activation section 26 is formed as a printing termination means, and when the print data which consist of a series of spool files sent from a client 10 are not able to receive to normal to the last, it stops printing processing.

[0027] In addition, in the above-mentioned configuration, spool files 11 and 22 are created by the disk unit (not shown) connected to each calculating machines 10 and 20, and the printing program 1 which are other configurations, spoolers 12 and 23, and the spool file reclosing section 13 will be created by software, and will demonstrate each function on CPU of each calculating machines 10 and 20.

[0028] It being characteristic in the gestalt of this operation is having transmitted the spool file 11 to the server 20 side, whenever it divided the print data created by the client 10 side for every specified quantity, it created the spool file 11 and the spool file's 11 was created. That is, it is having enabled it to perform creation processing of a spool file 11 based on a

printing data stream, and transfer processing to the server 20 of the created spool file 11 in parallel in a client 10. Thereby, the time amount which printing processing takes through the whole can be substantially shortened now.

[0029] Next, the printing processing in the gestalt of this operation is explained. The processing in the case of first, print data being created newly, and the print data being normally transmitted to a server 20, and being printed from a printer 21 is explained. In addition, drawing 3 is drawing having shown the timing diagram when printing processing is completed normally, and corresponds to drawing 17 of the conventional example.

[0030] First, as shown in drawing 4, creation processing of print data will be started (step 101), and the printing program 1 will create a printing data stream, if the printer 21 connected to the server 20 used as an output destination change and its server 20 is specified (step 102). And sequential sending out of the created printing data stream is carried out at a spooler 12 (step 103).

[0031] Drawing 5 is the flow chart which showed the printing processing which a spooler 12 performs. According to drawing 5, a spooler 12 is in the waiting state of generating of an event (step 111). Here, if a printing data stream is sent by having performed the printing program 1 (step 112), creation/transfer processing of a spool file 11 will be performed (step 113). The detail of creation/transfer processing of this spool file 11 is explained using the flow chart shown in drawing 6.

[0032] First, spool file creation / transfer routine which performs this processing is a routine called when a printing data stream occurs. First, it initializes by substituting [Ret] variables i and 1 (step 121). The spool file creation section 15 will be read into a buffer (not shown) with the magnitude of the specified quantity if a printing data stream is received serially (step 122). And the print data read for buffer size are changed into the predetermined format in which network transmission is possible, and it creates as a spool file 11 and saves (step 123). When this one spool file 11 is created, check processing which was shown in drawing 7 and which can be transmitted is performed (step 124). In addition, in the gestalt of this operation, although the specified quantity in case a printing data stream is divided will be dependent on a buffer size, it may set up the specified quantity of the arbitration which is not dependent on buffer size in consideration of transfer size, or a multi-buffer may be used for it. Setting out of the specified quantity is a mere design matter.

[0033] It judges whether the check routine which can be transmitted can be transmitted by referring to the flag of the server information table 17 which can be transmitted (step 131). With the gestalt of this operation, the value read in the server information table 17 is returned to spool file creation / transfer routine as it is (step 132,133).

[0034] In spool file creation / transfer processing shown in drawing 6, when it is in the condition (flag = which can be transmitted "0") which can transmit a spool file 11, the created spool file 11 is transmitted to a server 20 (step 125). Variable i is incremented when a transfer is completed normally (step 126).

[0035] On the other hand, when it is not in the condition which can transmit a spool file 11 (flag = which can be transmitted "1"), the server Research and Data Processing Department 18 is made to set "1" to the value of the flag corresponding to the server 20 of the destination of the server information table 17 which can be transmitted (step 127). And "0" which expresses

abnormalities to the return status Ret of this routine is substituted (step 128), and Variable i is incremented (step 126). In addition, Variable i is a variable for computing the number of partitions of a printing data stream, i.e., the total of a spool file 11 created, so that clearly from the above processing.

[0036] After the above-mentioned processing, when the read in to a buffer has not reached to the end of a printing data stream, it returns to the read in processing (step 122) to the buffer of a printing data stream (step 129). When the end of a printing data stream is reached, (i-1) is substituted for a global variable ALL, and the return status Ret is returned (step 130). In addition, the value of a global variable ALL is the total of the spool file created by dividing a printing data stream.

[0037] According to the gestalt of this operation as mentioned above, in spool file creation / transfer processing (step 113) shown in drawing 5, whenever a spool file 11 is created based on the created printing data stream, it will transmit to a server 20 serially. That is, since the transfer processing to a server 20 is started only by waiting only for the time amount by which a buffer is filled with the gestalt of this operation at a part of printing data stream, and a spool file 11 is created so that clearly from drawing 3, the printing processing time can be shortened substantially.

[0038] In drawing 5, although return (step 111) is carried out to the waiting state of an event, and a spool file 11 is deleted when it terminates normally (Ret=1) when transfer/creation processing of a spool file 11 terminates abnormally (Ret=0) (step 114), the spool file deletion which the spool file cutout 16 in this step 114 performs is explained using the flow chart shown in drawing 8.

[0039] In a spool file remove routine, after initializing Variable i by 0 (step 141), a spool file (i) is deleted (step 142). The above-mentioned processing is repeated and is performed until it increments Variable i (step 143) and Variable i reaches a global variable ALL (step 144). Thereby, all the spool files 11 created in the client 10 can be deleted.

[0040] Next, the printing processing in the server 20 to which a spool file 11 is transmitted is explained using drawing 9. First, it explains from the processing in always [when a series of spool files are receivable from a client 10 / forward].

[0041] The spooler 23 in a server 20 saves it as a spool file 22 by the side of a server 20, if a spool file is received (step 151). And a spooler 23 will start printing of the print data contained in the spool file 22, if a printer 21 is in the waiting state waiting for a printing demand when the first spool file is received (step 152). If it is in conditions other than the waiting for a printing demand, scheduling of the output processing to a printer 21 will be carried out. And the above-mentioned processing is repeated and is performed until the last spool file is transmitted from a client 10 (step 154). After reception and a printer 21 end printing for the spool file of last [spooler / 23] normally, that is notified to a client 10 (step 155).

[0042] By the way, although one printing data stream is divided into two or more spool files 11 and he is trying to transmit to a server 20 with the gestalt of this operation, the information showing the existence or nonexistence of the spool file 11 which follows is included in each spool file 11. Therefore, the spooler 23 of a server 20 can judge whether it is the existence or nonexistence and the tail end of a spool file which follow by checking this information at every reception of a spool file. When the spool file which originally follows and should be transmitted in step 152 does not

receive in predetermined time by this, printing will be stopped noting that a certain communication failure occurs (step 156). Here, the detail of the processing which stops printing which the printing control activation section 26 performs is explained using drawing 10.

[0043] The printing control activation section 26 investigates whether other spool files 22 which constitute the same printing data stream as the spool file which was not able to receive are printed at the time of generating of a time-out (step 161). If there is a thing under printing, a printing termination code will be sent out to a printer 21, and printing will be stopped compulsorily (step 162). On the other hand, if there is nothing under printing, it will judge that other already received spool files 22 are not printed yet, and will delete from a schedule (step 163). Then, a series of spool files 22 which constitute one printing data stream are deleted altogether (step 164). It is because it is useless even if it holds a series of spool files 22 received normally for abnormal termination. However, if the serial number the serial number is given to a series of spool files 22, and I want you to resend to a client 10 from a server 20 is notified, it is more efficient not to delete the spool file [finishing / reception] 22. Thus, also when a failure occurs in the communication link between client-server, it can be coped with in a server 20.

[0044] With the gestalt of this operation, whenever the printing data stream of the specified quantity was created as mentioned above, the spool file was sent to the server 20. If drawing 3 is compared with drawing 17, it is quite obvious that not only the job by the side of a client 10 but the time amount which printing processing takes has been shortened substantially. Of course, the time amount which creation of print data, data transfer, reception, and each printing take is the same as the conventional example. In the gestalt of this operation, since it was made to perform the creation processing and transfer processing of a spool file in a client 10 by dividing print data and creating a spool file in parallel, the time amount which printing processing takes can be shortened substantially.

[0045] Although the above processing explained processing in case print data are normally transmitted to a server 20 and are fundamentally printed from a printer 21 next, it explains the processing at the time of abnormalities when the created spool file 11 is not transmitted to a server 20. In addition, drawing 11 is drawing having shown the system time chart when detecting abnormalities to the transfer processing between client-server, and corresponds to drawing 18 of the conventional example.

[0046] In drawing 5, if a printing data stream is sent when a spooler 12 is in the waiting state of generating of an event, creation/transfer processing of a spool will be performed (steps 111-113). The detail of creation/transfer processing of this spool was already explained using the flow chart shown in drawing 6. By the way, although it judges whether the communication link with the appointed server 20 is possible by check processing which was shown in step 124 in drawing 6, i.e., drawing 7, and which can be transmitted, this check processing that can be transmitted is performed when the first spool file 11 is created. That is, in the former, when a part of printing data stream 11, i.e., the first spool file, is created and it is going to transmit in the gestalt of this operation to having judged whether the communication link with the server 20 which serves as the destination only after waits to create the whole printing data stream would be possible, the judgment can be made.

[0047] As mentioned above, when it is in the condition that the transfer to a server 20 cannot be performed (i.e., when the return status from spool

creation / transfer processing in step 113 is "0" showing abnormal termination), it returns to the waiting state of generating of an event, but (step 111) also while being unable to perform a transfer, creation processing of the remaining spool files 11 is performed continuously.

[0048] By the way, the server Research and Data Processing Department 18 will set to "0" the flag of the server information table 17 corresponding to the server 20 concerned which can be transmitted, if it detects that supervised the communication link condition between servers periodically oneself, or the failure was indirectly restored by the setting-out input by a system administrator etc. Although the flow chart of the processing in the spool file reclosing section 13 is shown in drawing 12, the spool file reclosing section 13 is started by the system administrator etc., or monitors continuously the flag of the server information table 17 which can be transmitted at the time of a failure, if it detects that the flag of the server 20 concerned which can be transmitted was set to "0", will recognize it as the communication link having been restored (step 171), and will carry out sequential sending out of the spool file 11 currently created by the spooler 12 (step 172).

[0049] In drawing 5, the spooler 12 which was carrying out waiting for an event will perform spool file transfer processing, if a spool file 11 is received from the spool file reclosing section 13 (step 111,112,115) (step 116). This processing becomes almost the same with having eliminated spool file creation processing from spool file creation / transfer processing shown in drawing 6. The flow chart of this spool file transfer processing is shown in drawing 13.

[0050] In drawing 13, a spooler 12 is first initialized by substituting [Ret] variables i and 1 (step 181). And the spool file 11 currently created is transmitted to a server 20 (step 182). Variable i is incremented when a transfer is completed normally (step 183). The above-mentioned processing is repeated and is performed until Variable i reaches a global variable ALL in this processing (step 184).

[0051] On the other hand, when a transfer of a spool file 11 goes wrong, the server Research and Data Processing Department 18 is made to set "1" to the value of the flag corresponding to the server 20 of the destination of the server information table 17 which can be transmitted (step 185). And "0" which expresses abnormalities to the return status Ret of this routine is substituted (step 186). And when a transfer of a spool file 11 can perform normally and reaches the tail of a printing data stream, when a transfer of a spool file 11 is not able to perform normally, the return status Ret is returned (step 187), and this processing is ended.

[0052] As mentioned above, according to the gestalt of this operation, since the printing data stream was saved at the spool file 11, the spool file 11 created even if it did not reboot the printing program 1, when it restored from the failure can be transmitted to a server 20. And spool file deletion (step 114) mentioned above is performed after the spool file transfer processing (step 116) in drawing 5.

[0053] Since he is trying to start the transfer processing to a server 20 after one completion of creation of a spool file 11, the abnormalities of the communication link between client-server are detectable with the gestalt of this operation at the event, so that clearly from drawing 11 thereby -- the spool file 11 from a printing data stream -- the rehabilitation work of a failure can be immediately begun from that time, without waiting for no creation. Moreover, since a part for the time amount and the transfer processing to a server 20 can also be resumed early, the time amount which

printing processing takes can be shortened substantially.

[0054] By the way, although it is as having mentioned above, that the server Research and Data Processing Department 18 changes the set point of the server information table 17 When the processing performed in this server Research and Data Processing Department 18 is briefly explained using the flow chart shown in drawing 14 , the server Research and Data Processing Department 18 If Server Name and the value of the flag which should be updated and which can be transmitted are received (step 191), the appointed flag value which can be transmitted will be set to the flag of a server 20 with which the server information table 17 was specified and which can be transmitted (step 192). Thus, since it set it as the server information table 17 whether the transfer to each server to which the client 10 is connected would be possible, the judgment can be made easily.

[0055] Moreover, with the gestalt of this operation, it is also setting to have formed the output destination change change section 19 to one of the descriptions. The processing in this output destination change change section 19 is explained using the flow chart shown in drawing 15 .

[0056] The output destination change change section 19 will receive the value of the flag corresponding to the server 20 of the server information table 17 which can be transmitted from the server information table 17, if it receives (step 201), the output destination change, i.e., Server Name, of a spool file 11, (step 202). When the value is possible (flag = which can be transmitted "0") in a transfer, an output destination change is turned to a server 20 (step 203). When it is improper (flag = which can be transmitted "1") in a transfer, a printer 14 turns an output destination change to the local port where it connects (step 204). Thereby, even if the output from the printer 21 connected to the server 20 is impossible and it is at the time, print data can be made to output from the printer 14 alternative to having connected locally.

[0057] In addition, although the output was changed to the unconditionally local printer 14 with the gestalt of this operation when the transfer to a server 20 was impossible, it may not be the local printer 14 or you may make it change to other preliminary servers. Moreover, an output destination change can be changed to the server of arbitration by having the table which set up the change place.

[0058]

[Effect of the Invention] According to this invention, whenever it divided the print data created by the client side for every specified quantity, it created the spool file and the spool file was created, the spool file was transmitted to the server side. That is, since the transfer processing to a server is started and it enabled it to perform creation processing of a spool file based on print data, and transfer processing to the server of the created spool file in parallel after that when waiting only for the time amount by which a spool file is created, the time amount which the whole printing processing takes can be shortened substantially.

[0059] Moreover, since the print-data deletion means was established, the spool file which the transfer ended normally can be deleted.

[0060] Moreover, since printing was immediately started when the spool file transmitted to a server side was received, reception of a spool file and output processing to a printer can be performed in parallel. Thereby, the time amount which the whole printing processing takes can be shortened substantially.

[0061] Moreover, since it was made to delete from a schedule when the spool file which should be transmitted to a server side was normally unreceivable, and other spool files which constitute the same printing data stream as the

spool file were in the middle of printing, and it was stopped and was not printed yet, it is not necessary to carry out useless printing.

[0062] Moreover, after a failure's occurring and carrying out the reinstatement by having established the print-data reclosing means, a transfer of a spool file [being created and saved] can be resumed.

[0063] Moreover, a server information management means is established, and since it set it as the server information maintenance means whether the transfer to each server to which the client is connected would be possible, an improper judgment which can be transmitted can be made easily.

[0064] Moreover, since the destination change means was established, even if it is the case where the communication link with the server which should print essentially cannot be performed normally, the output destination change can be changed to other servers and local printers.

[Translation done.]